Particle Identification for HERMES Run I

Jürgen Wendland

November 29, 2001

Abstract: Identification of leptons and hadrons is crucial for any analysis of HERMES data. This note gives an update on the latest knowledge of particle identification (PID) at HERMES. The PID scheme applies a Bayesian algorithm to separate hadrons and leptons. The necessary probability distributions (parent distributions) are extracted with the *xparent* package, the best knowledge on PID is implemented in the user callable library *PIDlib*.

1 Introduction

In every physics analysis of HERMES data it is necessary to separate hadrons and leptons in the data sample. The HERMES spectrometer includes four detectors for particle identification: a lead glass calorimeter, a preshower detector, a transition radiation detector (TRD) and a threshold Čerenkov detector which was replaced by a ring imaging Čerenkov detector (RICH) in 1998.

At HERMES, particles are identified using a probabilistic algorithm that utilizes the responses of these four detectors. The resulting logarithmic ratio of the hadron and lepton probability is commonly called *PID*. The first section briefly describes the formalism to calculate the PID values for each track by means of Bayesian statistics.[1] In the following section the available tools, namely a PID library and a stand-alone code to generate necessary input distributions, are introduced and documented.

2 PID Formalism

Identification of hadrons and leptons is based on the responses of the four PID detectors in the spectrometer, that are different for the two particle types. The responses are combined into probabilities using a Bayesian algorithm. In this way all available information enters into the particle identification and thus the separation of hadrons and leptons is maximized. The benefit of the probabilistic approach is illustrated in figure 1.

The goal of the algorithm is to calculate the probabilities $P(H_{l(h)}|E,p)$ for the hypothesis $H_{l(h)}$ that the track is a lepton (hadron) given a track momentum p and an energy deposition E in the chosen detector. By means of Bayes' Theorem, this conditional probability may be written as

$$P(H_{l(h)}|E,p) = \frac{P(H_{l(h)}|p)\; P(E|H_{l(h)},p)}{P(E|p)}. \qquad (1)$$

It is thus necessary to calculate the probabilities $P(E|H_{l(h)},p)$ that a lepton (hadron) with momentum p will deposit an energy E in the detector and the *prior probability* $P(H_{l(h)}|p)$ that a track with momentum p is a lepton (hadron). The probability $P(E|p)$ that a particle with momentum p will

[1] Detailed discussions of the particle identification scheme are found in [2, 3, 4, 5]

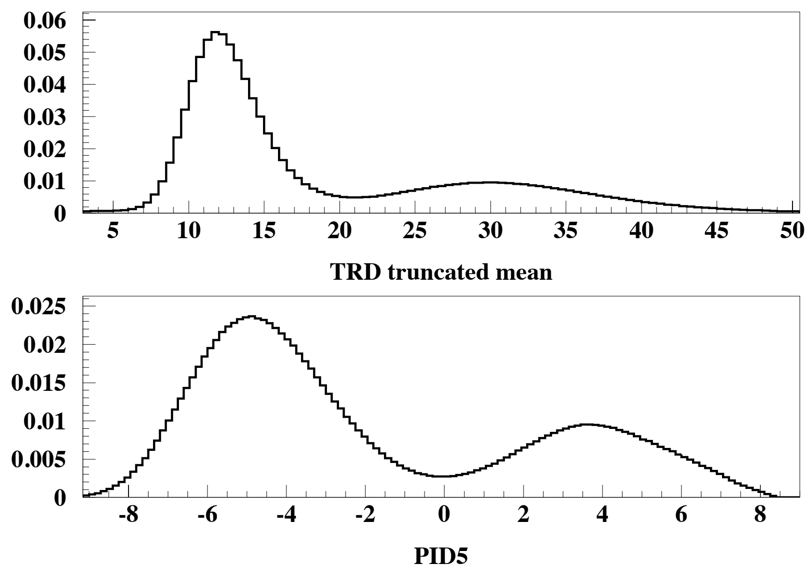

Figure 1: The top plot shows the TRD truncated mean distribution. The graph illustrates the separation of hadrons (left peak) from the broad lepton peak at high energies. The lower plot shows the flux corrected PID5 distribution of the same data sample. The hadron peak at negative PID values is obviously better separated from the lepton peak at positive PID values.

deposit the energy E in the detector is given by

$$P(E|p) = \sum_{i=l,h} P(H_i|p)\; P(E|H_i, p),$$

hence

$$P(H_{l(h)}|E, p) = \frac{P(H_{l(h)}|p)\; P(E|H_{l(h)}, p)}{\sum_{i=l,h} P(H_i|p)\; P(E|H_i, p)}. \tag{2}$$

The probability distributions $P(E|H_{l(h)}, p)$, commonly called *parent distribution*, for example may be measured at test-beam facilities with appropriate lepton and hadron beams. At HERMES, however, a different approach is used. The parent distributions are extracted from real data by imposing cuts on the PID detectors other than the one considered. For example to generate

lepton parent distributions for the TRD, cuts are imposed on the responses of the calorimeter and the preshower detector. This procedure assumes that the responses of the detectors for a given particle type are uncorrelated. Generating clean particle samples has the advantage that the parent distributions take into account aging effects and varying detector conditions during data taking periods. The generation of the parent distributions is described in more detail in section 3.1.

The prior probability $P(H_{l(h)}|p)$ is equivalent to the *lepton (hadron) flux* $\phi_{l(h)}$. These flux factors are calculated in an iterative procedure discussed in section 3.2.

Once obtained, the probabilities $P(H_l|E,p)$ and $P(H_h|E,p)$ are combined into a logarithmic ratio PID′,

$$\mathrm{PID}' = \log_{10} \frac{P(H_l|E,p)}{P(H_h|E,p)}. \tag{3}$$

In terms of the parent distributions and the flux factors this quantity becomes

$$\mathrm{PID}' = \log_{10} \frac{P(E|H_l,p)\; P(H_l|p)}{P(E|H_h,p)\; P(H_h|p)} = \mathrm{PID} - \log_{10} \Phi, \tag{4}$$

where

$$\mathrm{PID} \equiv \frac{P(E|H_l,p)}{P(E|H_h,p)} \qquad \text{and} \qquad \Phi \equiv \frac{\phi_h}{\phi_l} = \frac{P(H_h|p)}{P(H_l|p)}. \tag{5}$$

At HERMES the following combinations of PID values are commonly defined,

$$\begin{aligned} \mathrm{PID2} &\equiv \mathrm{PID}_{cal} + \mathrm{PID}_{pre} \\ \mathrm{PID3} &\equiv \mathrm{PID}_{cal} + \mathrm{PID}_{pre} + \mathrm{PID}_{cer} \\ \mathrm{PID5} &\equiv \mathrm{PID}_{trd} = \sum_{i=1}^{6} \mathrm{PID}_{trd,i}, \end{aligned} \tag{6}$$

where the sum in the last equation runs over the six TRD modules.

In practice the flux ratio Φ is often neglected and only the quantity PID3 + PID5 (PID2 + PID5 for data taken after 1997) is used for particle identification. This approximation is valid as long as the flux ratio is fairly constant and/or sufficiently hard cuts on PID3 + PID5 are used. From

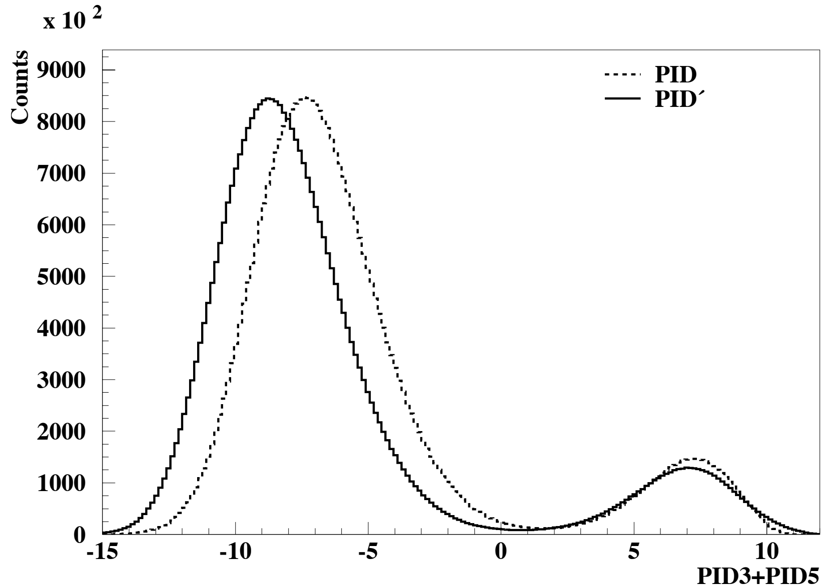

Figure 2: PID distributions with and without flux factors. The plot illustrates the improvement in discrimination of leptons (peak on the right hand side) and hadrons (peak on the left hand side) when PID3 + PID5 are flux corrected. The data are from the 97c1 production.

a Bayesian point of view the complete neglect of the information about the particle fluxes is not desirable. The difference between the quantities PID and PID$'$ is presented in figure 2. The plot illustrates that an analysis, where cuts at PID3+PID5 < 0 and PID3+PID5 > 2 are applied to identify hadrons and leptons respectively, is not very sensitive to the application of flux factors. However, it is obvious that due to the neglect of the flux information, the efficiency of the PID scheme is worse.

3 Generating Parent Distributions and Flux Factors with *xparent*

The main purpose of the program *xparent* is to extract parent distributions for the various data productions. It is also possible to carry out basic particle

identification analysis and to calculate particle fluxes.

The program processes HERMES data in the form of μdsts. The data have to pass the relevant data quality criteria[2] in order to be included in the analysis [6]. In addition only trigger 21 events are accepted in the analysis. Finally tracks are discarded, if they originate from vertices outside the target cell or if they are not fully inside the acceptance of the HERMES spectrometer.

3.1 Parent Distributions

		1996/ 1997	1998 - 2000
Calorimeter	leptons	$0.92 < E/p < 1.10$	$0.92 < E/p < 1.05$
	hadrons	$0.01 < E/p < 0.80$	$0.01 < E/p < 0.50$
Preshower	leptons	$0.025 < E$	$0.03 < E$
GeV	hadrons	$0.0 < E < 0.004$	$0.0 < E < 0.003$
TRD	leptons	$26.0 < E$	$26.0 < E$
keV	hadrons	$0.1 < E < 14.0$	$0.1 < E < 13.0$

Table 1: Cuts to identify leptons and hadrons for generating parent distributions. In the case of the calorimeter the cut is placed on the ratio of deposited energy E and track momentum p.

The parent distributions for each detector and particle type are the normalised count rates as a function of the detector response[3]. The count rates are extracted by imposing cuts to identify hadrons and leptons respectively on the PID detectors other than the one considered. The cuts were chosen such that the particle samples are sufficiently clean while ensuring a reasonable number of tracks in each parent distribution. Thus the cuts are restrictive for the large 1998 to 2000 datasets, whereas fairly loose cuts are used for the smaller 1996 and 1997 datasets. The cuts are summarised in table 3.1. Note that there are no cuts on the threshold Čerenkov or the RICH. It was found that the Čerenkov cuts for hadrons bias the hadron sample. In particular above the pion threshold the hadron sample selected by the Čerenkov detector is not representative, because pions emit Čerenkov radiation and are

[2]In the case of the 96 and 97 data, all bursts with any of the bad-bits $421E13DC$ are discarded. For 98, 99, and 2000 data the mask is $425E13DC$.

[3]In the case of the calorimeter the ratio of response E and track momentum p is used.

discarded from the sample by the cut. Thus the Čerenkov response is not part of the selection criteria for the parent distributions of the calorimeter, the preshower detector, and the TRD.

The parent distributions are arranged in a number of momentum intervals for each detector. The intervals were chosen such that the response of the detector within the interval is reasonably constant. At the same time the number of intervals was chosen as small as possible in order to minimize required memory. The momentum bins are listed in table 2.

For each momentum bin and particle type the parent distributions are generated as a function of the deposited energy, in case of the calorimeter as a function of the ratio of deposited energy and track momentum. The distributions are divided into 200 bins. The chosen energy intervals are listed in table 3. The detector responses for leptons and hadrons integrated over all momenta are shown in figure 3. The plots illustrate the capability of the four detectors to separate leptons and hadrons by their unique responses.

3.2 Flux Factors

It is possible to use the code *xparent* for the calculation of particle fluxes, once parent distributions are available. The particle flux ratio is a function of the momentum p and the polar angle θ, $\Phi \equiv \Phi(p, \theta)$, because the cross section depends on these quantities. In terms of the number of incident particles the flux ratio may be written as

$$\Phi(p, \theta) = \frac{(\text{number of incident hadrons } (p, \theta))}{(\text{number of incident leptons } (p, \theta))}. \tag{7}$$

In practice the flux ratio is calculated in a number of bins in momentum and polar angle. The bins are given in table 4. The flux ratio cannot be calculated exactly, because particle identification, a necessary input, depends directly on this ratio (see equation (4)). Instead it is estimated in an iterative procedure [3]. Initially the flux ratio is taken to be uniform,

$$\Phi_0(p, \theta) \equiv 1, \tag{8}$$

which essentially corresponds to uniform distributions of the probabilities to detect a hadron or a lepton. The data are then analysed and the flux ratios

$$\Phi_1(p, \theta) = \frac{(\text{number of tracks with } PID' < 0 \ (p, \theta))}{(\text{number of tracks with } PID' > 0 \ (p, \theta))} \tag{9}$$

Beam Energy 27.5 GeV, Calorimeter threshold 1.4 GeV			
		Bins	Intervals [GeV]
Calor.	lept.	6	0., 2.5, 3.5, 4.5, 6.5, 8.5, 30.0
	hadr.	8	0., 1.0, 1.5, 2.0, 4.0, 5.0, 7.0, 10.0, 30.0
Presh.	lept.	9	0., 1.5, 2.0, 2.5, 3.0, 4.0, 6.0, 8.5, 12.0, 30.0
	hadr.	4	0., 2.5, 6.0, 9.0, 30.0
Čer.	lept.	4	0., 4.4, 10.0, 17.0, 30.0
	hadr.	10	0., 1.0, 4.0, 4.4, 5.0, 6.0, 8.0, 10.0, 12.5, 15.0, 30.0
TRD	lept.	5	0., 6.0, 12.0, 18.0, 23.0, 30.0
	hadr.	6	0., 2.0, 4.0, 6.0, 8.0, 10.0, 30.0
Beam Energy 27.5 GeV, Calorimeter threshold 3.5 GeV			
		Bins	Intervals [GeV]
Calor.	lept.	6	0., 2.0, 3.0, 5.0, 7.0, 10.0, 30.0
	hadr.	13	0., 1.0, 1.5, 2.0, 3.0, 3.5, 4.0, 4.5, 5.0, 6.0, 7.0, 8.0, 9.0, 30.0
Presh.	lept.	9	0., 1.5, 2.0, 2.5, 3.0, 4.0, 6.0, 8.5, 12.0, 30.0
	hadr.	3	0., 3.0, 8.5, 30.0
Čer.	lept.	4	0., 4.4, 10.0, 17.0, 30.0
	hadr.	10	0., 1.0, 4.0, 4.4, 5.0, 6.0, 8.0, 10.0, 12.5, 15.0, 30.0
TRD	lept.	5	0., 6.0, 12.0, 18.0, 23.0, 30.0
	hadr.	6	0., 2.0, 4.0, 6.0, 8.0, 10.0, 30.0
Beam Energy 12.0 GeV, Calorimeter threshold 1.4 GeV			
		Bins	Intervals [GeV]
Calor.	lept.	6	0., 2.5, 3.5, 4.5, 6.5, 8.5, 15.0
	hadr.	8	0., 1.0, 1.5, 2.0, 4.0, 5.0, 7.0, 10.0, 15.0
Presh.	lept.	8	0., 1.5, 2.0, 2.5, 3.0, 4.0, 6.0, 8.5, 15.0
	hadr.	4	0., 2.5, 6.0, 9.0, 15.0
Čer.	lept.		
	hadr.		
TRD	lept.	4	0., 3.0, 6.0, 9.0, 15.0
	hadr.	6	0., 1.0, 2.0, 3.0, 4.0, 5.0, 15.0

Table 2: Momentum binning of the parent distributions. There are no Čerenkov momentum bins defined for the low beam energy data, because the data were taken with the RICH.

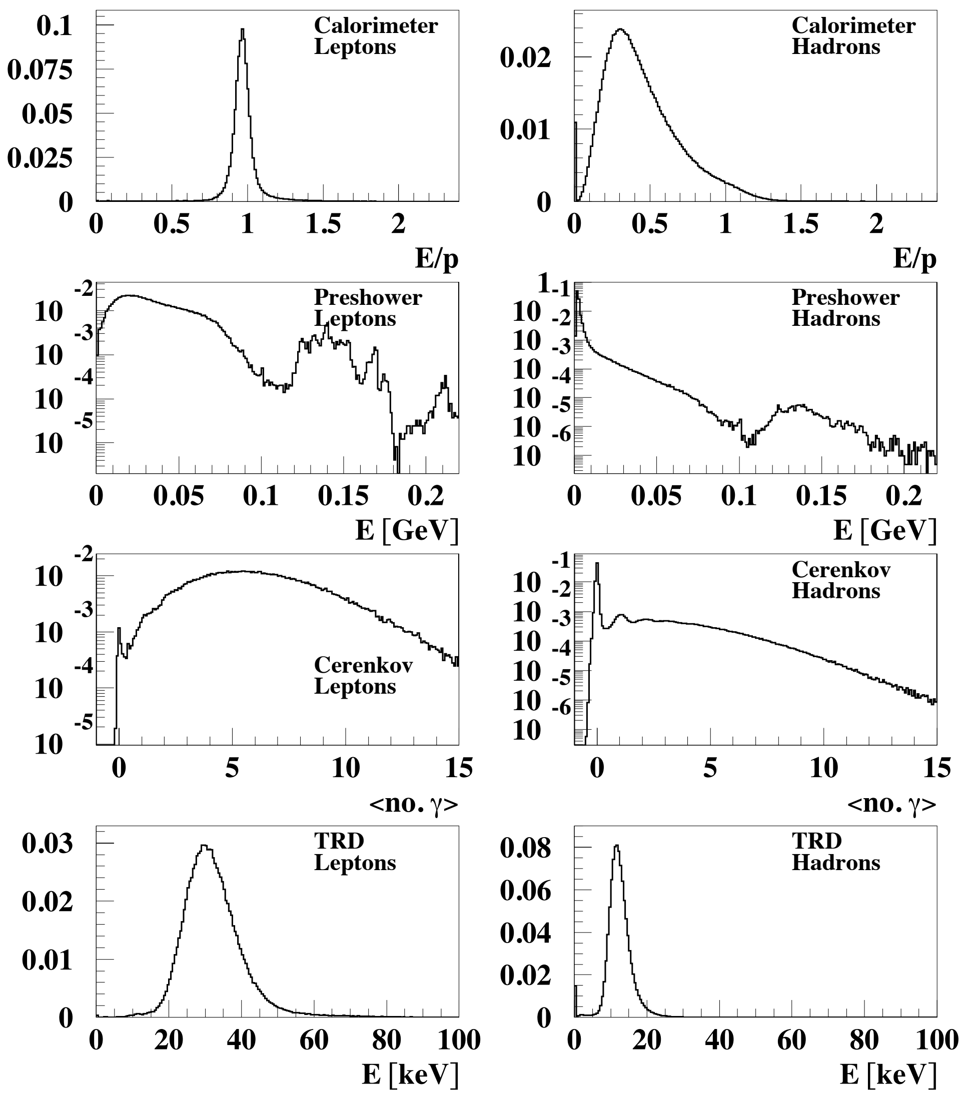

Figure 3: The PID detector responses for leptons and hadrons integrated over all momenta. The plots show the differences in the responses of the two particle types, which makes their distinction possible.

Detector	Lower Bound	Upper Bound
Calorimeter	0.	2.4
Preshower	0. GeV	0.22 GeV
Čerenkov	−1	15
TRD	0. keV	100. keV

Table 3: The energy ranges of the parent distributions. In case of the calorimeter the range is given in terms of the ratio E/p. In case of the Čerenkov the range is given in terms of the mean number of absorbed photons. Due to calibrations this number can be negative.

	Bins	Ranges
p	27	0., 1.4, 2.4, 3., 4., 5., 6., 7., 8., 9., 10., 11., 12., 13., 14., 15., 16., 17., 18., 19., 20., 21., 22., 23., 24., 25., 26., 30.
θ	6	0.040, 0.075, 0.100, 0.125, 0.150, 0.175, 0.300

Table 4: The momentum and polar angle bins for the calculation of the flux ratio

are calculated for each p-θ-bin. These ratios are used in the next iterative step to calculate PID' and subsequently the flux ratio Φ_2. The iteration is continued until convergence (see section 3.3.2 for an example).

3.3 Using *xparent*

3.3.1 Installation

The package is available on *http://hermes.desy.de/PID*. The web page also contains the latest information about the program. After downloading and extracting the gzipped tar archive using

```
tar xvzf xparent.tgz,
```

the code is easily compiled using the commands

```
cd xparent/source
./configure --use-environment
make
```

Note that the package requires a few CERN and HERMES libraries as well as the library *PIDlib*. Installation of the latter is described below (section 4).

3.3.2 Running the Program

The program is mainly driven through command line arguments. Executed without arguments the following small help text is provided:

```
[neelix ~]> xparent
syntax: xparent options
options:
  -p <prod>
  -t <threshold:  14 or 35>   : calo thresh
  -udst|-piddst|-ntuple|-mcdst: filetype of the runfiles
  -rl <runlist>|-fl <filelist>: runlist|filelist
  [-o <output directory>]     : default ./scratch
  [-bl <burstlist>]           : analyse  these bursts only
  [-bin <binning-file>        : def: <year>-thresh<thresh>.bins
  [-beam <beam energy>]       : default 27.570
  [-TRDcorr]                  : apply TRD x y correction
  [-Hi <PDhistofile>]         : create PDs from these histograms
  [-PID]                      : PID mode
  [-PD <directory containing PDs>]
  [-PIDntu]                   : output ntuple (PID mode only)
  [-flux [<fluxfile>]]        :
```

The only required options to run the code are -p <prod>, -t <thresh>, -udst|-piddst|-ntuple|-mcdst, and -rl <runlist>|-fl <filelist>:

-p prod: This option specifies the data production, it is mainly responsible for applying the correct data quality cuts.

-t <thresh>: This option specifies the calorimeter threshold. Supported thresholds are currently 1.4 GeV and 3.5 GeV, selected by '14' and '35' respectively.

-udst|-piddst|-ntuple|-mcdst: One of these options tells the code which file-type to expect. Selecting -udst specifies a μdst production which is probably correct for most data analysis. The options -piddst and

-ntuple are essentially only kept for historical reasons, enabling the code to analyse piddsts or ntuples. More importantly -mcdst specifies the analysis of fully tracked MC data.

-rl <runlist>|-fl <filelist>: This options specifies a run- or filelist in a file. A runlist is simply a list of run numbers followed by the fill number, e.g.

```
15533    -2
15534 25
15535 25
15536 25.
```

The fill number is arbitrary, negative fill numbers cause the code to skip the run. A filelist is simply a list of the full filenames of the runs including the path, e.g.

```
/home/hermes/opa/udstprod/udst_97c1/smlinks/run15533.smdst.gz
/home/hermes/opa/udstprod/udst_97c1/smlinks/run15534.smdst.gz
/home/hermes/opa/udstprod/udst_97c1/smlinks/run15535.smdst.gz
/home/hermes/opa/udstprod/udst_97c1/smlinks/run15536.smdst.gz.
```

The following options are not essential to run the program, but change its default behaviour:

-o <output directory>: This option selects the output directory. By default *xparent* writes its output to the directory ./scratch. Note that the program creates the subdirectories cal_parent, pre_parent, cer_parent, and trd_parent. These can also be created with the script mkPDdirs.sh which is part of the tar archive.

-bl <burstlist>: This option specifies that only bursts listed in the specified file are used for analysis. The bursts are still subject to data quality. The format corresponds to that of the first four columns of the official burstlist: fill, run, burst, μdst counter, e.g.

```
  24  15533  13231    1    4
  24  15533  13232    2    4
  24  15533  13233    3    4
  25  15534   1667    1    8
  25  15534   1668    2    8
```

-bin <binning-file>: This option selects the file containing information about the response and momentum bins (see tables 2 and 3). By default the filename is constructed from the year the data were taken and the calorimeter threshold: <year>-thresh<thresh>.bins. The format of the file is strictly fixed, e.g. for the calorimeter:

```
###Calo
200 0. 2.4
#leptons
6
2.5
3.5
4.5
6.5
8.5
30.0
#hadrons
[...]
```

'###Calo' selects the calorimeter, the following line specifies that the response is to be binned in 200 bins, from 0. to 2.4. Then the lepton momentum bins are specified. In this example there are 6 bins with the upper boundaries at 2.5 GeV, 3.5 GeV, etc. starting at 0. GeV. Note that the code is very sensitive to the format. The file must begin with the calorimeter, followed by preshower detector, the Čerenkov detector, and the TRD. Also the 'comments' must look exactly as shown in the files supplied in the archive.

-beam <beam energy>: This option selects the beam energy. The default is 27.57 GeV. Bursts with a significantly different beam energy are discarded. This option is applicable for 2000 data only.

-TRDcorr: This option specifies to use the TRD x and y correction (see section 5)

-Hi <PDhistofile>: Using this option causes the code to skip all data analysis. Instead a previously generated histogram file is read. The parent distributions in the form of histograms are extracted and output in the ascii files that are used for PID analysis. This option overrides any of

the settings driving data analysis. Thus the above mentioned required options may be omitted.

-PID: This option causes the code to run in *PID mode*. Instead of generating parent distributions, a basic PID analysis is carried out and PID histograms are output in the file `pid.histos.rz`. Note that in this mode only good polarised bursts are accepted in the analysis.

-PD <directory containing PDs>: (PID mode only) This option specifies the directory that contains the parent distributions to be used for PID analysis. The structure of the directory must be of the form:

```
directory
|-- th14
|   |-- cal_parent
|   |-- cer_parent
|   |-- pre_parent
|   `-- trd_parent
`-- th35
    |-- cal_parent
    |-- cer_parent
    |-- pre_parent
    `-- trd_parent
```

The directory containing parent distributions for the calorimeter threshold other than the one studied is not required. Apart from the parent distributions the directories `th14` and `th35` must contain the files `bins.14.txt` and `bins.35.txt` respectively with the binning information. These files are output by the program along with the parent distributions.

-PIDntu: (PID mode only) This option causes the code to accumulate an ntuple with relevant PID quantities.

-flux [<fluxfile>]: (PID mode only) This option includes the generation of particle fluxes. If no file is specified, the program will assume uniform fluxes as input. An iteration as described above is possible by running the program several times, with the fluxfile output read in the following run.

Note that the flux factors generally depend on the data sample used. The code *only* calculates flux factors for polarised analysis. For other types of analyses it is possible to adapt the source code and extract fluxes specific for that analysis. Typically the data-quality word defined in `main.F` (section "`c... define badbitmask`") and the geometry cuts hardcoded in the file `xparent_cuts.F` (subroutine `def_cuts`) need to be adjusted.

The program outputs several files, depending on the running mode selected. A list of these files is given in the following.

`parent.histos.rz`: This histogram file contains histograms necessary for the generation of the parent distributions. In addition a few standard histograms are accumulated. These show quantities like the particle momenta and their detector responses.

`cal_parent, pre_parent, cer_parent, trd_parent`: These directories contain the parent distributions for the four PID detectors. There is one file for each detector, momentum bin, and particle. Each file contains the corresponding parent distribution in the response bin as specified in the binning file.

As mentioned above, the directories must exist and may be created with the script `mkPDdirs.sh`.

`pid.histos.rz`: (PID mode only) This file contains histograms of the relevant PID distributions. In addition the same standard histograms as in `parent.histos.rz` are accumulated.

`pid.ntuple.rz`: (PID mode only) This file contains the ntuple that is generated when the option `-PIDntu` is used.

`flux.all.txt`: (PID mode only) This file contains the particle flux ratios for positive, negative, and all particles in a self-explanatory ASCII format. In the header of the file the χ^2-value of the comparison with the previous flux is given, too.

Example (i): Parent distributions for the 97c1 production.

In this case, *xparent* is called in the following way:

```
[worf xparent]> xparent -p 97c1 -t 14 -udst -fl 97c1filelist \
> -o results/97c1
```

Example (ii): PID analysis of the 97c1 production.

Here it is necessary to run the program in PID mode and to specify which parent distributions are to be used:

```
[worf xparent]> xparent -p 97c1 -t 14 -udst -fl 97c1filelist \
> -o results/97c1/th14 -PID -PD ./results/97c1/
```

An ntuple is generated, if `-PIDntu` is specified in addition. This is only advisable, if a small number of runs is analysed.

Example (iii): Particle Fluxes for the 97c1 production.

It is again necessary to run the program in PID mode. In order to carry out the iteration described above, the program has to be run several times. In the first run *xparent* is called without specifying a flux file:

```
[worf xparent]> xparent -p 97c1 -t 14 -udst -fl 97c1filelist \
> -o results/97c1/th14 -PID -PD ./results/97c1/ -flux
```

The second time *xparent* is run, the flux file output by the previous cycle is specified:

```
[worf xparent]> xparent -p 97c1 -t 14 -udst -fl 97c1filelist \
> -o results/97c1/th14 -PID -PD ./results/97c1/ \
> -flux results/97c1/th14/flux.all.txt
```

The program is rerun in this way until the difference between the flux factors from the current and the previous cycle becomes small. This convergence may be monitored by the χ^2 comparison of the previous and the current flux factors given in the header of the file `flux.all.txt`. The iteration usually converges after three to four cycles with final χ^2 values of the order of 10^{-2}.

4 The library *PIDlib*

The PID library *PIDlib* uses the parent distributions generated with *xparent*. The PID values for a given track are calculated using the track momentum

and detector responses in the algorithm described above (section 2). The library is handled in a straight forward way; an initialisation routine is called at the beginning of the data analysis, and PID values for a given track are calculated by a second routine.

4.1 Installation

The PID library is installed on the PC-farm in `/data02/juergenw/PIDLIB`. A separate installation on the farm is thus usually not necessary.

The library is available on *http://hermes.desy.de/PID* for a private installation. The latest information on changes and updates is also available on the web page. The gzipped tar archive extracts to the directory `./PIDLIB` using

```
tar xvzf PIDLIB.tgz,
```

and the code is easily compiled with the commands

```
cd PIDLIB/srcLIB
./configure --use-environment
make
```

The compiled library is called `./PIDLIB/libPID.a`. The flag

```
-L [...]/PIDLIB/ -lPID
```

has to be added to the Makefile of an analysis code in order to link it into the executable.

4.2 Using the Library in an Analysis

In an analysis code that uses the library, it is first necessary to initialise the parent distributions. The routine `PIDLIB_init` reads and stores the selected set of parent distributions in the corresponding arrays. The routine expects the directory from which to read the parent distributions as argument[4]:

```
call PIDLIB_init(<directory>)
```

The directory must have the following structure:

[4] The syntax given here is in FORTRAN. The routines may also be called from c, see appendix A

```
directory
|-- th14
|   |-- cal_parent
|   |-- cer_parent
|   |-- pre_parent
|   `-- trd_parent
`-- th35
    |-- cal_parent
    |-- cer_parent
    |-- pre_parent
    `-- trd_parent
```

Parent distributions for the standard data productions are available in `./PIDLIB/PDs`. Note that the routine will output warnings if some files are not available. These warnings may be ignored, if e.g. data with calorimeter threshold of 1.4 GeV are studied, and the warnings only concern parent distributions for the 3.5 GeV threshold. The routine terminates the calling program, if crucial files are not found.

The PID values for a given track are calculated by the routine `PIDLIB_PID`. Note that this routine returns the quantities PID_{cal}, PID_{pre}, PID_{cer}, and PID_{trd}. It does *not* incorporate any flux factors. The routine expects nine arguments; a typical call has the form

```
call PIDLIB_PID(thresh,p,phi,calo,pre,cer,trd,PID,ierr).
```

The arguments are:

`thresh`: (Input) The calorimeter threshold. The library supports the thresholds 1.4 and 3.5 GeV, specified by 1.4 and 3.5, respectively. In the μdst productions the threshold is stored in `g1Quality_rCaloThresh`.

`p`: (Input) The particle momentum (`g1Track_rEnergy`).

`phi`: (Input) The azimuthal angle ϕ. This input is currently not in use. Future versions of the library may use separate parent distributions for the top and bottom detector halves (`g1Track_rPhi`).

`calo`: (Input) The energy deposited in the calorimeter by the partial back track (`smTrack_rECaloBackPT`).

`pre`: (Input) The energy deposited in the preshower detector (`smTrack_rPre`).

cer: (Input) The mean number of Čerenkov photons (`smTrack_rCer`).

trd: (Input) The six TRD module responses (`smTrack_rTRDPulsRaw(6)`).

PID: (Output) The PID values for the four detectors:

- **PID(1):** Calorimeter PID
- **PID(2):** Preshower PID
- **PID(3):** Čerenkov PID
- **PID(4):** TRD PID.

ierr: (Output) Error flag: 1 – threshold unknown, 0 – otherwise.

The PID values returned by the subroutine are related to the standard PID values through

$$\mathrm{PID2} = \mathrm{PID}(1) + \mathrm{PID}(2)$$

$$\mathrm{PID3} = \mathrm{PID}(1) + \mathrm{PID}(2) + \mathrm{PID}(3)$$

$$\mathrm{PID5} = \mathrm{PID}(4).$$

In the following analysis these values may then be used in the same way as the PID values stored in the μdst files.

5 The TRD x and y Correction

The *xparent* package and the PID library also incorporate the TRD x and y correction described in [7]. This algorithm corrects a spatial dependence of the TRD bottom detector, that affected its overall performance. The correction is implemented in the routine `PIDLIB_TRD_xycorr`. A typical call is

```
call PIDLIB_TRD_xycorr(phi,x,xslope,y,yslope,TRD,TMEAN)
```

where the arguments are:

phi: (Input) The azimuthal angle of the track (`g1Track_rPhi`).

x: (Input) The x position of the back partial track at the center of the magnet (`smTrack_rXpos`)

xslope: (Input) The x slope of the back partial track (`smTrack_rXSlope`).

y: (Input) The y position of the back partial track at the center of the magnet (`smTrack_rYpos`).

yslope: (Input) The y slope of the back partial track (`smTrack_rYSlope`).

TRD: (Input/ Output) The six TRD module responses (`smTrack_rTRDPulsRaw(6)`). The uncorrected values will be replaced by the corrected values.

TMEAN: (Output) The corrected TRD truncated mean response (`g1Track_rTRD`).

It is imperative that the corrected TRD module responses be used in combination with corrected parent distributions to calculate the x and y corrected PID value PID5. It is thus necessary to load parent distributions that include the correction, to correct the TRD module responses with the routine `PIDLIB_TRD_xycorr` and then to calculate the PID values with the routine `PIDLIB_PID`. Corrected parent distributions are available for most of the data productions, the directories have the suffix `.TRDxycorr`. Note that the correction should not be used with uncorrected parent distributions.

A Using the PID library in a c Code

The library can be called from a c code in a straight forward way. In the declarations it is necessary to include the following definitions:

```
#include "cfortran.h"

#define PIDLIB_INIT(A) \
  CCALLSFSUB1(PIDLIB_INIT,pidlib_init,PSTRING, A)

#define PIDLIB_PID(A1,A2,A3,A4,A5,A6,A7,A8,A9) \
  CCALLSFSUB9(PIDLIB_PID,pidlib_pid,PFLOAT,PFLOAT,PFLOAT \
              ,PFLOAT,PFLOAT,PFLOAT,FLOATV,FLOATV,PINT \
              ,A1,A2,A3,A4,A5,A6,A7,A8,A9)

#define PIDLIB_TRD_XYCORR(A1,A2,A3,A4,A5,A6,A7) \
```

```
  CCALLSFSUB7(PIDLIB_TRD_XYCORR,pidlib_trd_xycorr,PFLOAT \
             ,PFLOAT,PFLOAT,PFLOAT,PFLOAT,FLOATV,PFLOAT \
             ,A1,A2,A3,A4,A5,A6,A7)
```

The three routines PIDLIB_INIT, PIDLIB_PID, and PIDLIB_TRD_XYCORR can now be called as shown in the following example:

```
main() {
  char directory[] = "./PDs/99b2";

  float thresh, p, phi, calo, pre, cer;
  float trd[6];
  float PID[4];
  int ierr;

  float x, xslope, y, yslope, tmean;

  PIDLIB_INIT(directory);

  PIDLIB_PID(thresh, p, phi, calo, pre, cer, trd, PID, ierr);

  PIDLIB_TRD_XYCORR(phi, x, xslope, y, yslope, trd, tmean);

}
```

B Sample Output of *xparent*

As mentioned in section 3.3.2 the program *xparent* generates parent distributions by default. In this mode the program starts up with the message

```
 ****************************************
  xparent version 2.0 (2001/11/21)
     Extracting parent distributions.
 ****************************************
```

Upon finishing analysis, the program outputs some statistics and terminates:

```
Bursts with threshold 1.4:             6820
Bursts with threshold 3.5:                0
```

```
Good Bursts with selected thresh:       6820
Tracks viewed:     957755
Good Tracks:       461126

Output directory: scratch/96c2/


xparent is ending normally
```

The output files are the file `bins.14.txt` that contains information on the response and momentum bins used in the analysis, the file `test.dat` that can be used to output debugging information, and the histogram file `parent.histos.rz`. The latter file contains a number of histograms; the first few are listed below.

```
PAW > h/fil 33 parent.histos.rz
PAW > h/lis

 ===> Directory :
         3 (1)   random number
         6 (1)   all bursts: HERA Beam Energy
         7 (1)   acc bursts: HERA Beam Energy
         8 (1)   all bursts: HCalorimeter threshold
         9 (1)   acc bursts: Calorimeter threshold
        10 (1)   Momentum
        20 (1)   Theta
        30 (1)   Phi
       100 (1)   Calorimeter e/p
       110 (1)   Preshower
       120 (1)   Cerenkov
       130 (1)   TRD truncated mean
       610 (1)   TRD module 1
       620 (1)   TRD module 2
       630 (1)   TRD module 3
       640 (1)   TRD module 4
       650 (1)   TRD module 5
       660 (1)   TRD module 6
[...]
```

C Paw Macros

The *xparent* archive includes two paw macros in the subdirectory `kumac/`. The macro `comp_parent.kumac` may be used to compare two sets of parent distributions. The macro reads histogram files output by *xparent* and generates plots for each parent distribution, that may be used to compare the two datasets. Executed without arguments the following help text is given:

```
syntax: exec comp_parent thresh file1 file2 outpsfile
        thresh       -- 14 or 35, the calorimeter threshold x 10
        file1,file2  -- histogram files with PDs to be compared
        outpsfile    -- postscript output file, will be zipped.
```

The second macro, `check_DSTpid.kumac` may be used to compare the PID values stored in the μdsts with those calculated with the PID library. The macro reads the ntuple `pid.ntuple.rz` output by *xparent* in PID mode when the flag `-PIDntu` is used. The full path to this file has to be supplied as argument when the macro is executed.

D Parent Distributions in the μdsts

The parent distributions that are used in the μdst-writer differ, because of two approximations made in the case of the TRD. The hadron response of the TRD is virtually identical for the six modules. Thus the μdst-writer uses only one set of hadron parent distributions for the six modules. Secondly parent distributions for the TRD are averaged over the two calorimeter thresholds, because the TRD response does not depend on this threshold.

There are two shell scripts provided in the *xparent* archive in the subdirectory `scripts/`. The script `averageTRDhad.sh` may be used to average the TRD hadron parent distributions over the modules. Executed without arguments a short message is printed describing the syntax. Two sets of parent distributions may be averaged by means of the shell script `averagePD.sh`. The syntax is also printed, if executed without arguments. The script calls a short perl macro `calc_ave.pl` that is provided.

The parent distributions for the μdst-writer are organised in a different directory structure. For example in case of the parent distributions for data taken in 1996, the directories are named as follows:

```
|-- CALO_96_14
|-- CALO_96_35
|-- CERE_96_14
|-- CERE_96_35
|-- PRES_96_14
|-- PRES_96_35
`-- TRD_96
```

Currently there does not exist a shell script that arranges the parent distributions in this way.

References

[1] *HERMES Particle Identification*, http://hermes.desy.de/PID

[2] J. Wendland, Master's thesis, Simon Fraser University, (1999), HERMES internal note 99-016

[3] F. Menden, Diploma Thesis, TRIUMF/ Hamburg, (1997), HERMES internal note 98-001

[4] R. Kaiser, *Particle Identification at HERMES*, (1997), HERMES internal note 97-025

[5] R. Kaiser, Ph.D. thesis, Simon Fraser University, (1997), HERMES internal note 97-017

[6] *HERMES Data Quality*, http://hermes.desy.de/~oma/dataquality

[7] M. Hartig, *The TRD x and y correction*, (2001), in preparation